AN INVESTMENT-BASED APPROACH FOR MANAGING SOFTWARE-INTENSIVE SYSTEMS

Margaret E. Myers

Maintaining information superiority will be vital to the 21st-century warfighter, and the military's documented shortcomings in acquiring leading-edge information technology systems must be addressed in order to meet this need. The investment-based approach to the acquisition of software-intensive systems discussed here considers recent management reform legislation and will help DoD meet information superiority requirements.

In spite of numerous studies documenting the problems encountered in the acquisition of software-intensive systems, the defense acquisition community has not fully implemented the recommendations from those studies. As a result, the acquisition problems persist. Yet today's national security environment demands even more flexibility and responsiveness from the defense acquisition process, with software-intensive systems often on the leading edge of both the Revolution in Military Affairs and the Revolution in Business Affairs. This article recasts some of the historical recommendations in the light of recent management reform legislation and describes an investment-based management approach to the acquisition of software-intensive systems. Although the concepts described here are applicable to both hardware and software development, the scope of this article is limited to the management of systems with extensive software components, to include command and control systems, automated information systems, and other information technology investments.

WHY CHANGE IS NEEDED

The document *Joint Vision 2010* (Chairman of the Joint Chiefs of Staff, 1996) describes the future direction of our joint warfighting forces based on the emerging operational concepts of dominant maneuver, precision engagement, focused logistics, and full-dimension protection. Execution of these concepts depends on our ability to achieve and maintain information superiority (CJCS, 1996):

> Sustaining the responsive, high-quality data processing and information needed for joint military operations will require more than just an edge over an adversary. We must have information superiority: the capability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same.

The Department of Defense (DoD) Acquisition Year 2000 goal (Gore, 1997) of delivering new major defense systems to the users in 25 percent less time is especially relevant to implementation of Joint Vision 2010, which depends heavily on DoD's ability to leverage new and emerging technological opportunities. Unfortunately, the department's track record in keeping up with the rapid pace of advances in commercial information technology (IT) is not good, and many software-intensive systems fail to achieve their key performance parameters.

Although defense acquisition policy has evolved from the time when major defense acquisition programs were mostly hardware, the acquisition process still often requires extensive tailoring for software-intensive systems. However, very little guidance is available on how to tailor the policy for these systems. (See Appendix A for descriptions of various acquisition and software development models.) A different approach is needed for software-intensive systems, which must keep pace with technological advances while being responsive to the warfighter.

Implementing the management approach described here will support information superiority requirements by delivering software-intensive systems that are more responsive to the needs of the 21st century warfighter.

PROPOSED MANAGEMENT APPROACH

The following recommendations are based on an analysis of various acquisition and development models, legislation, policy guidance, and best practices relevant to software-intensive systems. The recommendations focus primarily on changes to the management and oversight processes since the technical implementation will, of necessity, vary from system to system.

ADOPT AN INVESTMENT FOCUS

For most acquisition programs, success is defined in terms of gaining Milestone III approval to produce and deploy the system, which is essentially a one-time event. A more appropriate perspective for software-intensive systems may be to view them as evolving capital assets that will provide a needed capability for some number of years. For software-intensive systems, that capability will be delivered incrementally to the user over the life of the investment. The key is to develop a long-term investment focus in support of goals that span the life of the program, not just to deliver a one-time product and walk away. This capital asset perspective is consistent with the Government Performance and Results Act (1993) and Office of Management and Budget (OMB) capital planning guidance. (For more information on GPRA and the OMB Capital Planning Guide [OMB, 1997], see Appendix B.)

Define Investment Objectives

For DoD systems, the value of a capital asset should be measured in terms of its contribution toward achieving one or more goals in the DoD strategic plan (currently the Quadrennial Defense Review [QDR]) (Cohen, 1997). Given the proposed "evolving capital asset" perspective described above, the requirements and acquisition communities should jointly develop intermediate investment objectives that are acceptable to the user and technically feasible. The acquirer subsequently translates these objectives into capability packages that, when deployed, demonstrate measurable progress toward meeting the DoD strategic goals. The system developer derives the specific technical requirements for each capability package based on the user's objectives. When deployed, each capability package should demonstrate measurable progress toward achieving the intermediate objectives and, ultimately, the strategic goals.

The key is for management to be able to maintain traceability from the Joint Vision 2010 concepts, to the DoD strategic plan and supporting strategic goals, to the investment objectives, and finally to the implementing capability packages. The challenge lies in defining investment objectives that are measurable and preferably quantifiable. The health affairs community is probably the leader in holding its management accountable by measuring progress against strategic goals and investment objectives. Defining system-level objectives and linking them to corporate strategic goals are key tenets from the GPRA, Clinger-Cohen Act of 1996, and OMB guidance. (See Appendix B for more information on the Clinger-Cohen Act.)

Build an Investment Framework

The decision to invest in a software-intensive capital asset should initiate planning for an investment framework (business model) to manage that asset during its useful life. This framework should include not only the operational and technical architectures that will define how the capital asset will be used and built, but also repeatable processes for updating the investment objectives, negotiating the scope of each increment, evolving the software components, managing the risks, and measuring the outcomes. For deeply embedded applications, a DoD-driven domain analysis and architecture are essential, with an emphasis on classic software reuse paradigms; for many information systems, a market-driven analysis and architecture that can leverage the commercial sector may be more appropriate. For command and control systems, a hybrid approach is usually required to deal with acquiring and integrating commercial-off-the-shelf (COTS) and government-off-the-shelf (GOTS) applications into custom-developed software. Some of the challenges for hybrid systems include modification of COTS packages to interoperate with custom-

"The challenge lies in defining investment objectives that are measurable and preferably quantifiable."

developed software; the resulting maintenance, licensing, and ownership issues; synchronization of changes with existing GOTS software that will continue to evolve independently; and ground rules for each increment to retain maximum flexibility for future design and requirements changes.

From a management and oversight perspective, building the investment framework to support the production of follow-on increments should be just as important as deploying the first increment. The investment framework is analogous to establishing a software production line to streamline the development of following increments; this approach was successfully demonstrated in the Software Technology for Adaptable, Reliable Systems (STARS) project sponsored by the Defense Advanced Research Project Agency (Institute for Defense Analysis, 1996). The concept of an investment framework is consistent with the Clinger-Cohen Act, which mandates an integrated technology architecture. (For more information on architectures for software-intensive systems, see Appendix C; for more information on software product lines, see Appendix A.)

"The goal should be to deliver small, compatible increments that provide useful, added capability every 6 to 18 months."

CONSTRAIN INCREMENT SIZE

A tenet of recent legislation and guidance is that information technology systems should "be implemented in phased, successive segments as narrow in scope and brief in duration as practicable, each of which solves a specific part of an overall mission problem and delivers a measurable net benefit independent of future segments" (Raines, 1996). One of the lessons learned from program managers who have implemented software-intensive systems based on the incremental or evolutionary models is that the first increment typically fails to meet its cost, schedule, and performance parameters because the scope is too broad. This usually happens because the user is unwilling to constrain the requirements because of fears that follow-on increments won't be delivered.

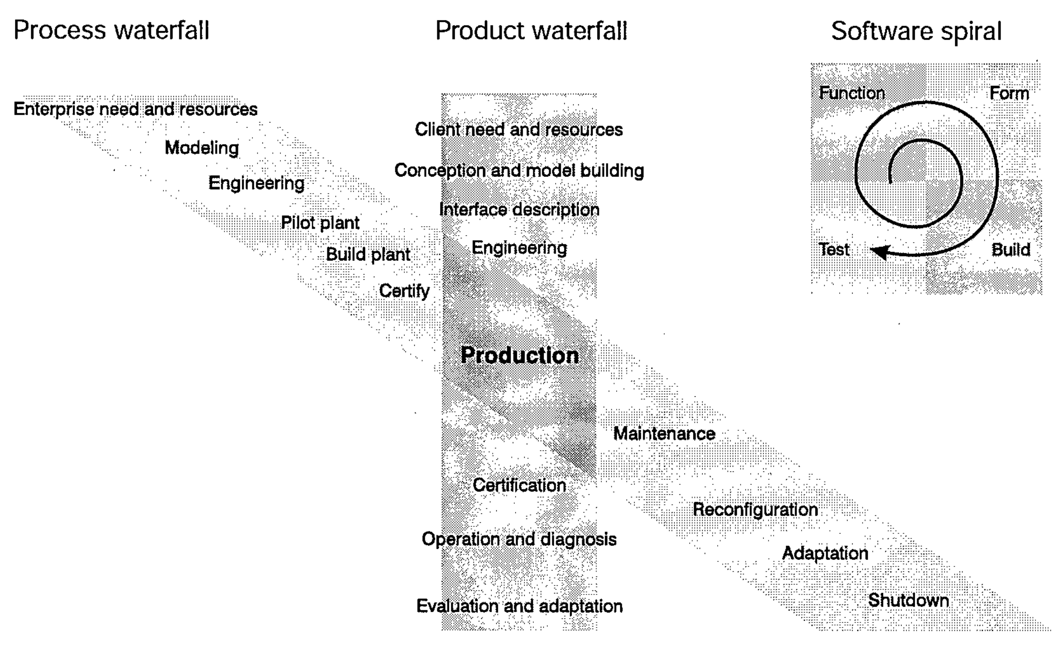
Adopting a capital asset perspective and constraining increment size should shift the focus from one of demanding full capability in the first increment to defining the minimum useful capability for the first and each subsequent increment. The goal should be to deliver small, compatible increments that provide useful, added capability every 6 to 18 months. The Global Command and Control System (GCCS), for example, is currently on an 18-month schedule for deploying major releases, with smaller beta releases in-between. The Army Tactical Command and Control System (ATCCS) currently plans to deploy new software increments approximately every 12 months. Smaller increments reduce risk, minimize schedule delays, and avoid cost overruns. This is consistent with the Clinger-Cohen Act and OMB guidance.

APPLY THE SPIRAL-TO-CIRCLE MODEL

Rechtin and Maier (1997) discuss the differences between the waterfall model, which aptly fits the largely irreversible steps of hardware acquisition, and the spiral model, which better represents the iterative process of software development (Figure 1). Although current defense acquisition policy strongly supports tailoring, most acquisition strategies resemble the waterfall model rather than the spiral model. After analyzing the structural dissimilarities between the two models and the problems that result when coordinating hardware and software development, Rechtin and Maier recommend use of a single spiral-to-circle model (Figure 2).

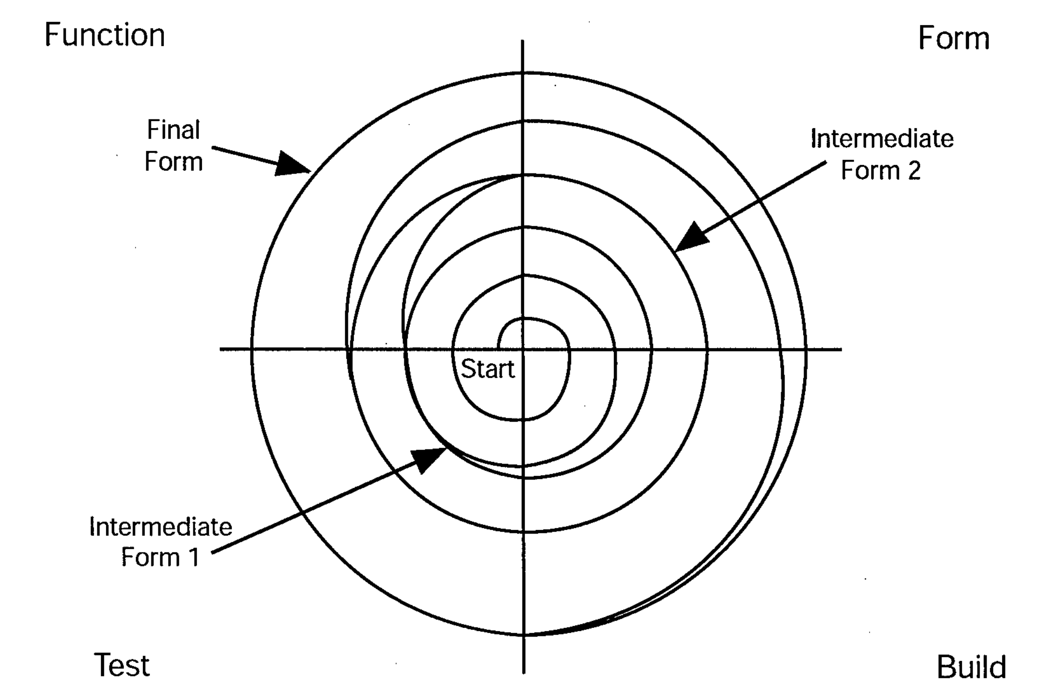
This model is based on the following heuristic: "Complex systems will develop and evolve within an overall architecture much more rapidly if there are stable intermediate forms than if there are not." For software development, the spiral-to-circle model implies pausing on the outward spiral by entering a closed circle for a stable version, which could be deployed and which would form the baseline for the next increment of functionality. For hardware development, the model implies a hold after each step to review progress. For combined hardware and software development, the closed circles represent the points at which stable hardware and software configurations come together for testing and potential deployment.

The spiral-to-circle model appears to be a useful management tool whenever it is necessary to integrate hardware and software components in the same system. The model is also applicable to hardware-intensive systems that are developed using simulation-based acquisition methodologies. Additionally, the model should be a



Source: *The Art of Systems Architecting*, by E. Rechtin and M. Maier, 1997, CRC Press.

Figure 1. Waterfall and Spiral Models



Source: *The Art of Systems Architecting,* by E. Rechtin and M. Maier, 1997, CRC Press.

Figure 2. The Spiral-to-Circle Model

useful integration tool when commercial items or other nondevelopmental items are used in lieu of developing new components. (For more information on the spiral-to-circle model, see Appendix A.)

INVESTMENT MANAGEMENT ISSUES

The investment-based approach just described (adopt an investment focus, define investment objectives, build an investment framework, constrain increment size, and apply the spiral-to-circle model) is intended to support information superiority requirements by delivering software-intensive systems that are more responsive to the needs of the 21st-century warfighter. To accomplish this, the approach must address three areas related to investment management of software-intensive capital assets. First, are the management issues associated with designing, developing, and deploying the core increment that will provide the initial operating capability? Second, are the issues associated with managing the follow-on increments? (These issues endure for the life of the system.) Third, are the interoperability issues that arise in coordinating the design, development, and deployment of increments from multiple systems (systems of systems)?

CORE INCREMENT ISSUES

Adopting the capital asset investment approach with its emphasis on up-front planning will require increased participation from the requirements (user) community, especially in defining the investment objectives and constraining increment size. One way to ease this burden would be to appoint an acquisition-qualified program manager to coordinate the planning activities before the investment is approved as an acquisition program. This, in turn, would require some additional training for the program manager and might conflict with current initiatives to reduce the size of the acquisition workforce.

Building the investment framework is not trivial. The GCCS evolutionary acquisition process appears cumbersome to those who see it for the first time, but it was invented by the GCCS integrated product team members (who received the Defense Acquisition Executive Award for Acquisition Excellence for their initiative), and it seems to work effectively for GCCS. Unless the investment framework processes for other programs are carefully established and the people are effectively trained, the software-intensive capital asset concept is no better than current acquisition approaches. (For additional information on GCCS, see Appendix C.)

Follow-on Increment Issues

Once the investment framework is effectively established and has been proven to work on the first increment, follow-on increment development should have lower risk, especially if the increments are schedule-constrained. The Milestone Decision Authority should consider delegating follow-on deployment decisions, but some limited oversight may be required to ensure that the process remains disciplined.

Software-intensive systems that have already deployed their core increment are candidates for conversion to the investment approach once they have established an appropriate investment framework, to include a current baseline. The GCCS evolutionary acquisition process, for example, was developed after the core increment was deployed.

Systems of Systems Issues

The investment framework must include a process for ensuring interoperability with other systems and increments from other software-intensive capital assets. This is especially critical in supporting the Joint Vision 2010 requirement for information superiority. The Army uses the spiral-to-circle model to address synchronization issues associated with the Army Battlefield Control System (ABCS). The ABCS component systems must successfully complete a synchronization event to demonstrate interoperability before deployment. Beta sites and test beds are also useful tools for validating interoperability before deployment. Constraining increment size should be conducive to scheduling synchronization events and establishing OT&E test windows, in which multiple systems have an opportunity to jointly test their newest increments before full deployment. (For more information on operational test and evaluation [OT&E] strategies for software-intensive systems, see Appendix C.)

"The investment framework must include a process for ensuring interoperability with other systems and increments from other software-intensive capital assets."

PROCESS CHANGES REQUIRED FOR IMPLEMENTATION

Implementing the investment-based approach described here will require acquisition, requirements, and PPBS process changes, to include changes in policy, guidance, and training.

ACQUISITION PROCESS

The recommendations suggested above are consistent with defense acquisition policy, which allows for extensive tailoring. However, the proposed approach should be documented in the *Defense Acquisition Deskbook* (DAD, 1998) as a DoD-wide best practice and updated with implementation lessons learned.

Implementing the concepts described above will not work without an investment in education and training for program managers, their staffs, and other personnel in the acquisition chain. Team training for the participants in each specific project may be the most efficient way to introduce these new concepts. Specific topics that must be addressed include the GPRA, the Clinger-Cohen Act, OMB capital asset guidance, architectures, and software management issues, to include the use of software process and product quality measures. The Software Engineering Institute's software capability maturity model (CMM) and software acquisition CMM are examples of models that can be used to promote the process improvements needed to build and manage an investment framework.

"The acquisition community must partner with the Joint Staff to jointly identify needed changes to the requirements process in support of the software-intensive capital asset approach."

REQUIREMENTS PROCESS

The acquisition community must partner with the Joint Staff to jointly identify needed changes to the requirements process in support of the software-intensive capital asset approach. One of the key lessons learned and relearned in the acquisition of software-intensive systems is the need to involve the real end user, both in helping to refine the specific requirements and in assessing how well those specific requirements, as they are implemented, meet their needs. The GCCS beta release strategy mentioned previously allows users to experiment with new applications on a trial basis; only those applications that the users want are incorporated into the next major release. The GCCS evolutionary acquisition strategy supports this flexible approach to requirements generation, but most major acquisition programs do not have this flexibility.

PLANNING, PROGRAMMING, AND BUDGETING SYSTEM (PPBS) PROCESS

The comptroller and the acquisition community should jointly identify needed changes to the PPBS process to support the software-intensive capital asset approach. To best implement the approach described here, program managers need a guarantee of program stability and a steady-state funding stream. The comptroller should also work with OMB to ensure that the proposed investment process is implemented consistently with OMB guidance.

Other Implementation Suggestions

In addition to integrating necessary changes into the acquisition, requirements, and PPBS processes, it may be necessary to charter a multifunctional process action team to develop the policy, guidance, and training required to implement the proposed approach. One or more pilot programs would be useful for maturing the new processes and demonstrating the improvement.

Conclusion

This investment-based approach to the acquisition of software-intensive systems meets information superiority requirements, while complying with recent management reform legislation. The proposed approach is based on five key recommendations: adopting an investment focus, defining investment objectives, building an investment framework, constraining increment size, and applying the spiral-to-circle model. The approach can be adapted to address issues related to core increments, follow-on increments, and systems of systems. Successful implementation will require coordinated changes to the acquisition, requirements, and PPBS processes and a better understanding of how to tailor acquisition strategies. These changes, however, are essential to delivering software-intensive systems that are more responsive to the needs of the 21st-century warfighter.

Margaret E. Myers is the director of performance assessment for the Office of the Assistant Secretary of Defense (OASD) for Command, Control, Communications, and Intelligence (C3I). She was previously the director of acquisition oversight for the OASD(C3I) and served as acting Deputy Assistant Secretary of Defense for C3I Acquisition. Other prior positions include deputy commander and technical director of the U.S. Army Operational Evaluation Command (OEC), director of information mission area systems evaluation (OEC), deputy director of command control, communications, and computer (C4) systems evaluation (OEC), and chief, Systems Performance and Special Projects Division for the Army Information Systems Engineering Command.

Her current responsibilities include independently assessing DoD information technology (IT) performance to ensure that investments are consistent with DoD policies and plans and provide measurable improvements; formulating DoD IT acquisition policy applicable to all DoD components; planning and coordinating the IT acquisition review process for the OASD (C3I); and coordinating the Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance (C4ISR) Cooperative Research Program.

She earned a B.A. degree in mathematics from Colorado College, an M.S. degree in operations research from American University, and a Ph.D. degree in information technology from George Mason University.
(E-mail address: margaret.myers@osd.pentagon.mil)

REFERENCES

Boehm, B. (1988, May). A spiral model of software development and enhancement. *Computer*. Vol. 21, No. 5. The Institute of Electrical and Electronics Engineers, Inc.

Chairman of the Joint Chiefs of Staff. (1996). *Joint vision 2010*. Washington, DC: GPO.

Clinger-Cohen Act of 1996, Pub. L. 104-208, 104th Congress. (Formerly the ITMRA.)

Cohen, W. S., U.S. Department of Defense (1997). *Report of the quadrennial defense review*. Washington, DC: GPO.

Defense acquisition deskbook (1998). [On-line and CD-ROM]. Available http://www.deskbook.osd.mil

Gore, A. (1997). *DoD acquisition reinvention impact center goals*. Washington, DC: GPO.

Government Performance and Results Act of 1993, Pub. L. 103-62, 103rd Congress.

Information Technology Management Reform Act of 1996 (ITMRA), Pub. L. 104-106, 104th Congress.

Institute for Defense Analysis. (1996). *The cost and benefits of domain-oriented software reuse: Evidence from the STARS demonstration projects* (IDA Paper P–3191). Alexandria, VA: Author.

Office of Management and Budget. (1997). *Capital planning guide* (Supplement to Circular A–11, Part 3). Washington, DC: GPO.

Raines, F. D. (1996). *Principles of budgeting for capital asset acquisitions.* Washington, DC: Office of Management and Budget.

Rechtin, E., & Maier, M. W. (1997). *The art of systems architecting*. Boca Raton, FL: CRC Press.

APPENDIX A

ACQUISITION AND SOFTWARE DEVELOPMENT MODELS

ACQUISITION PROGRAM STRUCTURE MODELS

The following information is extracted from the *Defense Acquisition Deskbook (1998)*. The program structure is the fundamental building block of the program's acquisition strategy, where "program structure" means the phases and milestone decision points established for a program. The program structure models described below, when appropriately tailored, are suitable for the vast majority of major programs. One of the major themes in the current version of the DoD 5000 policy is that Milestone Decision Authorities (MDAs) "should strive to tailor most aspects of the acquisition process, including program documentation, acquisition phases, and the timing, scope, and level of decision reviews."

Traditional model. This model is the four-milestone, four-phase process that represents the department's typical approach to major acquisition development programs. Because of its widespread use, statutory requirements tend to be associated with this model's phases and milestone decision points.

Grand design model. This model is characterized by acquisition, development, and deployment of the total operational capability in a single increment. The required operational capability can be clearly defined and further enhancement is not foreseen to be necessary. The grand design model is most appropriate when the user requirements are well understood, supported by precedent, easily defined, and assessment of other considerations (e.g., risks, funding, schedule, size of program, or early realization of benefits) indicates that a phased approach is not required.

Incremental model. The incremental model is generally characterized by acquisition, development, and deployment of capability through a number of clearly defined system increments that stand on their own. The number, size, and phasing of the increments required for satisfaction of the total scope of the stated user requirement should be defined by the program manager, in consultation with the user. An incremental model is most appropriate when the user requirements are well understood and easily defined, but assessment of other considerations (e.g., risks, funding, schedule, size of program, or early realization of benefits) indicates a phased approach is more prudent or beneficial. An example of this model is pre-planned product improvement.

Evolutionary model. This model is characterized by the design, development, and deployment of a preliminary capability using current technology that includes provisions for the evolutionary addition of future capabilities as requirements are further defined and technologies mature. The evolutionary model differs from the incremental model in that the total functional capability is not completely defined at inception, but evolves as the system is built. This model offers an alternative to the traditional model for those programs not requiring a leap in technology, where the design process includes technology maturation, and where a program can make use of an interim solution with successive upgrades.

Advanced concept technology demonstrations (ACTDs) and evolutionary models share some similarities in that both involve short cycle times and address a requirement for state-of-the-art technology. ACTDs, however, are oriented to the development of an operational concept and do not necessarily result in a production program. Evolutionary models are oriented toward production from the beginning. (Note: The *Defense Acquisition Deskbook* contains several excellent sources of additional information on the evolutionary model, including the DSMC *Guide for Evolutionary Acquisition,* the Australian Defence Department handbook, and the *Global Command and Control System Lessons Learned.*)

Other program models. The models described above may be tailored to support commercial item and nondevelopmental item acquisitions.

DEVELOPMENT MODELS

The following descriptions of the waterfall, spiral, and spiral-to-circle models are extracted from *The Art of Systems Architecting* by Rechtin and Maier (1997).

Waterfall model. The waterfall model describes a sequence of largely irreversible steps especially typical of hardware acquisition and production plant construction. Although the waterfall method is less appropriate for software development, it is sometimes used for software-intensive systems.

Spiral model. The iterative process of software development is better represented by a spiral expanding through four quadrants: function, form, build (code), and test. In the DoD environment, function equates to requirements definition; form equates to design; build equates to development; and test equates to test and evaluation. In this model continually expanding software versions are based on learning from earlier development.

The spiral model is attributed to Boehm (1988), who developed and applied the model to large government software projects while working for TRW. The spiral model creates a risk-driven approach to the software process, rather than primarily a document-driven or code-driven process. Each cycle of the spiral begins with the identification of the objectives of the portion of the product being elaborated (performance, functionality, ability to accommodate change, etc.); the alternative means of implementing this portion of the product (design A, design B, reuse, buy, etc.); and the constraints imposed on the application of the alternatives (cost, schedule, interfaces, etc.). The following steps evaluate the alternatives, and identify and resolve risks; develop and verify the next-level of product; and plan the next phases.

Spiral-to-circle model. This single-process model accommodates the imperatives of both the hardware and software development processes based on the following heuristic: Complex systems will develop and evolve within an overall architecture much more rapidly if there are stable intermediate forms than if there are not. In hardware development, the model implies scheduled holds at the end of each step in the sequence to review the development and to determine that the integrity of the system concept has not been violated (everything necessary has been done and nothing unnecessary has been added). In software development, the model implies pausing in the outward spiral from time to time by going into a closed circle to create a stable version.

Because the spiral-to-circle model is a single model, it implies that the intermediate form is not only stable, but could also usefully continue as a product indefinitely (even as an acceptable end point should budget constraints or operational needs so dictate). Meanwhile, research, development, analysis, prototyping could continue to cycle on that circle until the decision is made to expand outward to new functions and forms.

Software product line model. Software product lines are software systems that share a set of common attributes (e.g., functionality, architecture, design, components/modules, development/maintenance processes). With these common attributes as a foundation, unique systems can be built to satisfy specific customers' requirements. The product line model was prototyped by the Defense Advanced Research Projects Agency (DARPA) software technology for adaptable, reliable systems (STARS) program. The STARS pilots successfully demonstrated the benefit of developing a common architecture and standards within a software domain (i.e., command and control) and then exploiting that common base to significantly reduce the design, development, and testing time for follow-on applications in that domain. More information on the STARS project is available on the World Wide Web at http://www.asset.com/stars/. The Defense Information Infrastructure Common Operating Environment (DII COE) is a product line focused on the infrastructure (vice application) level.

APPENDIX B

GPRA, CLINGER-COHEN ACT, AND OMB IMPLEMENTING GUIDANCE

GOVERNMENT PERFORMANCE AND RESULTS ACT

The Government Performance and Results Act (GPRA) of 1993 required agencies to submit strategic plans to the Office of Management and Budget (OMB) by September 30, 1997. The plans were to include:

- a comprehensive mission statement for major functions and operations of the agency;
- general and outcome-related goals;
- a description of how the agency will achieve the goals and the operational processes and resources required;
- a description of how the goals relate to annual performance plan goals;
- an identification of key factors external to, and beyond the control of, the agency that could significantly affect the achievement of goals; and
- a description of program evaluations the agency used in establishing and revising general goals, with a schedule for future program evaluations.

The DoD Strategic Plan is the Quadrennial Defense Review.

CLINGER-COHEN ACT

The purpose of the Clinger-Cohen Act of 1996 is to improve the productivity, efficiency, and effectiveness of federal programs through the improved acquisition, use, and disposal of information technology (IT) resources. Among other provisions, the law requires executive agencies to design and implement a process for maximizing the value and assessing and managing the risks of IT acquisitions. The Clinger-Cohen Act also streamlines the IT acquisition process by encouraging the adoption of smaller, modular IT acquisition projects. With certain exceptions, the Clinger-Cohen Act is generally applicable to National Security Systems.

OMB CAPITAL PLANNING GUIDANCE

The OMB *Capital Planning Guide* (Supplement to Circular A–11, Part 3) integrates various asset management initiatives (GPRA, Clinger-Cohen Act, etc.) into a single, integrated capital planning process to ensure that capital assets contribute to the achievement of agency strategic goals and objectives. The definition of capital assets includes IT hardware, software, and modifications; and DoD weapons systems. The four phases of the capital planning process are planning, budgeting, procurement, and management-in-use.

In the planning phase, the intent is for strategic plans, annual performance plans, and plans for capital assets to flow from the same process for identifying a baseline of current performance and the gap between current and planned performance; functional requirements for bridging this gap; alternatives for meeting these functional requirements; the best capital asset solution if one is needed; and a summary of proposed funding, procurement, and management of each capital asset within the agency's portfolio of assets in an agency capital plan. The acquisition strategy and risks are part of the information provided when seeking approval of a project.

Although budgeting begins in the planning phase, the formal start of the budgeting phase is the agency's request to OMB for asset acquisition. Agency budget submissions should be consistent with the "Principles of Budgeting for Capital Asset Acquisitions," which was published with the fiscal year 1998 budget. DoD guidance for implementing these principles is documented in the May 1, 1997, Office of the Secretary of Defense memorandum, "Requirements for Compliance with Reform Legislation for Information Technology (IT) Acquisitions (Including National Security Systems)." The budgeting phase ends when Congress appropriates funds for the acquisition and OMB apportions the funds to the agency.

OMB's procurement phase is essentially equivalent to the DoD acquisition process. Key steps in this phase are to:

- validate the planning decision;
- manage the procurement risk;
- consider tools (modular contracting, two-phased acquisition, competitive prototyping);
- select contract type and pricing mechanism;
- issue the solicitation;
- conduct proposal evaluation and negotiation;
- award the contract;
- manage the contract;
- conduct acquisition analysis; and
- conclude with acceptance (testing).

The management-in-use phase includes the steps an agency should take to manage and evaluate the continued viability of an acquired capital asset as part of the agency portfolio. The steps in this phase include:

- operational analysis (which can be used to minimize the cost of asset ownership while simultaneously improving the function the asset performs);
- execution of the operation and maintenance plan;
- post-implementation review (to evaluate the overall effectiveness of the agency's capital planning and acquisition process); and
- execution of the asset disposal plan.

APPENDIX C

OTHER RELEVANT GUIDANCE AND BEST PRACTICES

ARCHITECTURE SYNCHRONIZATION

DoD has adopted the concept of multiple, linked architectures to describe the operational, system, and technical views of information technology-based systems. Comprehensive DoD-wide architectural guidance is described in the Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework Version 2.0, which was approved for implementation in February 1998. Version 2.0 of the C4ISR Architecture Framework is available at http://www.cisa.osd.mil.

The following architecture descriptions are from various DoD architecture Web pages.

Operational architecture. An operational architecture is a set of elements consisting of information exchange requirements, mission area interactions, tasks, interoperability tables, logical connectivity, and a description of the environment where the information system is to be operated. The operational architecture is tied to both the systems and technical architectures and provides a disciplined approach or methodology to review baseline requirements, assess doctrinal impacts, examine and assess alternatives through excursions (functional or process improvements; and doctrine, training, leader development, organization, materiel, and soldiers [DTLOMS] requirements). An operational architecture:

- identifies the mission objective;
- identifies information exchange requirements;
- identifies logical connectivities; and
- identifies operational elements.

Systems architecture. A systems architecture view is a description, including graphics, of systems and interconnections providing for or supporting warfighting functions. It is a representation that associates physical systems and their performance attributes to the operational architecture and is built following the standards in the technical architecture. A systems architecture:

- maps information exchange requirements;
- defines connections between components;
- defines capacity;
- defines performance; and
- defines constraints.

Technical architecture. A technical architecture is a minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements that together may be used to form a system, and whose purpose is to ensure that a conformant system satisfies a specified set of requirements. A technical architecture identifies the services, interfaces, standards, and their relationships. A technical architecture:

- defines systems rules;
- establishes standards for interoperability; and
- applies technology references that influence architecture decisions.

(*Note: The Joint Technical Architecture is mandatory for all C4I systems.*)

Flexible OT&E Strategies

OT&E strategy for software-intensive systems. Since 1992, the Army has used a flexible operational test and evaluation (OT&E) strategy to support faster fielding of software-intensive systems that have been divided into blocks of functionality (increments). The strategy allows partial fielding of software-intensive systems, once successful OT&E of a representative sample has been accomplished. A representative sample is the portion of the software to be developed that demonstrates the ability of the hardware, commercial off-the-shelf (COTS) software, and communications network to support the total system requirements. The strategy is applicable both to weapon systems with extensive embedded software and information systems. The approach supports multiple software development models, enhances the program manager's acquisition strategy, and reduces the risk to the warfighter and the decision maker.

OT&E guidelines for software-intensive system increments. In October 1996, the Office of the Director of Operational Test and Evaluation (DOT&E) published guidelines intended to streamline the OT&E process and to achieve "affordable confidence" for the development and procurement of software-intensive systems. The guidelines apply to increments of software-intensive systems acquired subsequent to deployment of the "core block," which undergoes full operational testing. For insignificant to moderate risk increments, these guidelines streamline the OT&E process by reducing the degree of testing. The guidelines are applicable to both the incremental and evolutionary models.

OT&E test windows. One of the issues that the 1989 Army Science Board Summer Study on the Army Tactical Command and Control System (ATCCS) addressed was how to synchronize changes to the component ATCCS programs after the core systems were deployed. The recommended solution was to establish operational test "windows" that would be scheduled once or twice a year so that developers could ensure continued interoperability and minimize operational risk before deploying follow-on increments. The Army Program Executive Office for Command, Control, and Communications Systems has recently proposed a similar process to synchronize the development, testing, and fielding cycles of the Army Battlefield Command System component systems.

Blocked ORDs

Users occasionally write operational requirements documents (ORDs) that divide the requirements into "blocks" for incremental design, development, and deployment, but there is currently no explicit guidance on how to "block" ORDs. Several years ago, the automated information systems community proposed an approach by which the user and program manager would work together to sectionalize the ORDs, relying on the user's operational (functional) knowledge and the program manager's technical knowledge. The premise was that a viable acquisition strategy requires an ORD that can be implemented both technically and operationally. If not done collaboratively, the user may propose a solution that is not technically viable; conversely, the program manager may propose a technical solution that cannot be implemented operationally. The proposal also included suggestions for defining system increments in terms of functionality, user class or echelon, or operational mode.

GLOBAL COMMAND AND CONTROL SYSTEM

The Global Command and Control System (GCCS) has implemented an evolutionary acquisition strategy that integrates the requirements and acquisition processes to ensure the early, concurrent consideration of operational, technical, procedural, test, support, and fiscal issues within the GCCS stakeholder community. The *Defense Acquisition Deskbook* has information on the GCCS evolutionary acquisition process. Additional information is contained in an Institute for Defense Analysis paper that describes how the integrated product team process and DoD 5000 series policy were tailored to accommodate the evolutionary nature of GCCS. The IDA paper is available on the Web at: http://www.ida.org/DIVISION/sfrd/IDA_Papers_Documents.html.

Acknowledgments

This work was supported by the Office of the Assistant Secretary of the Army for Research and Development.

I thank Herb Fallin for his guidance in this project and Walt Hollis for his support. I also thank the following individuals for their ideas and comments on the concepts and draft paper: Linda Brown, Bill Campbell, Hoss Cartwright, Dan Czelusniak, Ron Garant, Jerry Garretson, Rick Johnson, Boyd Jones, Bob Kern, Jim Linnehan, Scott Lucero, Fred Myers, Lloyd Pickering, Tom Purdue, Al Resnick, Pat Sanders, George Schneiter, Robert Schwenk, Doug Sizelove, and Cindy Trackwell.

Correspondence should be addressed to Margaret E. Myers, OASD(C3I), 6000 Defense Pentagon, Washington, DC 20301.